Predicting User Sentiments For Hotels

CAPSTONE PROJECT

COURSE: **APPLIED DATA SCIENCE CAPSTONE**

SUBMITTED BY : SARAFUDHEEN M THARAYIL

Business Problem

- You are hired as a data scientist by a restaurant chain (probably the most prestigious) to tune the business based on customer review. It is found that recently there is a decline of 5% in the revienue in the last quarter. The restaurant management want to analyze the current situation and see what the individual customer feel about the restaurant.
- The Restaurant management decided to give a bonus point for their customer if they give a review (good or bad) in foursquare website in their venue in the chain of business.
- As there are many outlets and many thousands of users for the restaurants looking at each comment and review is not possible. So, as a data scientist, you are asked to prepare a model to predict if the user is happy or unhappy based on the comments without reading each and every line in it.
- The management also want to see what are the main keyword in the review (positive and negative) based on which the management can take actions.

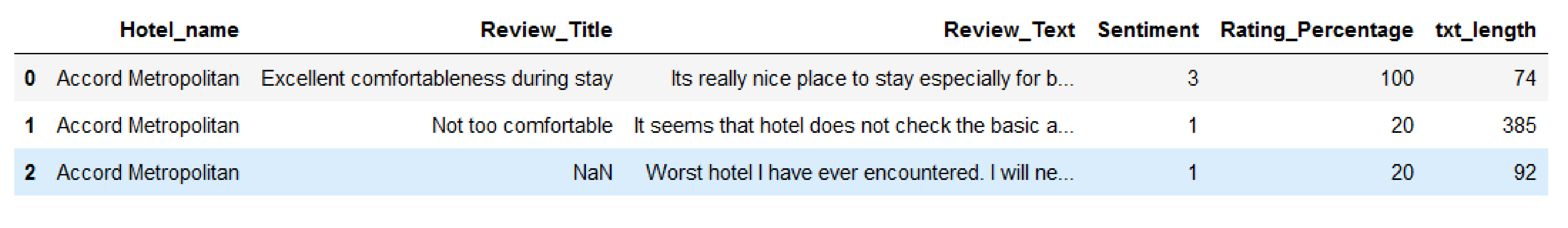
Overview of the solution

- A robust prediction model need to be prepared based on the available reviews and scroes that is available. The avaialable soruce for sentiments and its rating need to be reliable. Once we recive the data(in our case we got a **keggle dataset**), the data need to be analyzed and make sure that we have enough data to try with.
- Once we get the data, we do cleaning, remove stop words and vectorize it so that we can apply different machine learning algorithms. In our case, we use a classification algorithm to classify the comments from 1 - 3.
- Based on the words, we can find different words which can be identified as bad reviws and make some special care about those cases.
- Once we prepare the model, we could use the model to find the specific venue in the foursqure and find the recent user comments. Also check if the user comments are positive or negative. If they are negative, take special care and make necessary actions to rectify those bad reviews based on the keywords used in the review.
- We use sentiment analysis in scilearn to make a predictive model in order to see if the user review is positive, negative, or nutral.
- We will use the available sentiments from different restaurants to train the model. Based on the trained model, we will test the prediction based on sample test data. once the testing is found successful, we will use the new reviews from the user and give a score based on the trained model

Data Preparation Stage

- Initially we thought of getting the data from Foursquare, but it is found that the data for tips is premium category, so it has a limit to get the number of reviews.
- We have searched for different datasets for Indian context and found the data found a good one at https://www.kaggle.com/ranjitha1/hotel-reviews-city-chennai/version/2. This dataset is good for experimentation purpuse.
- We use this dataset and downloaded to 'chennai_reviews.csv' and we read data to dataframe

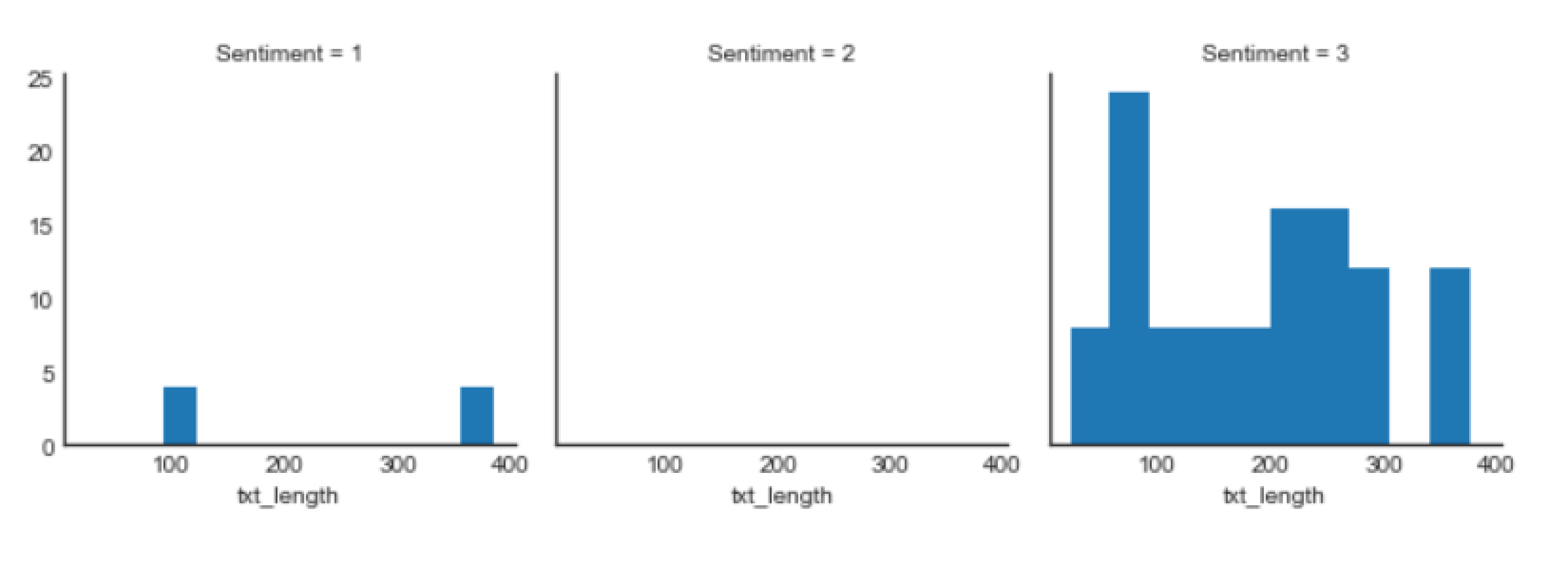
Sample Data Gathered



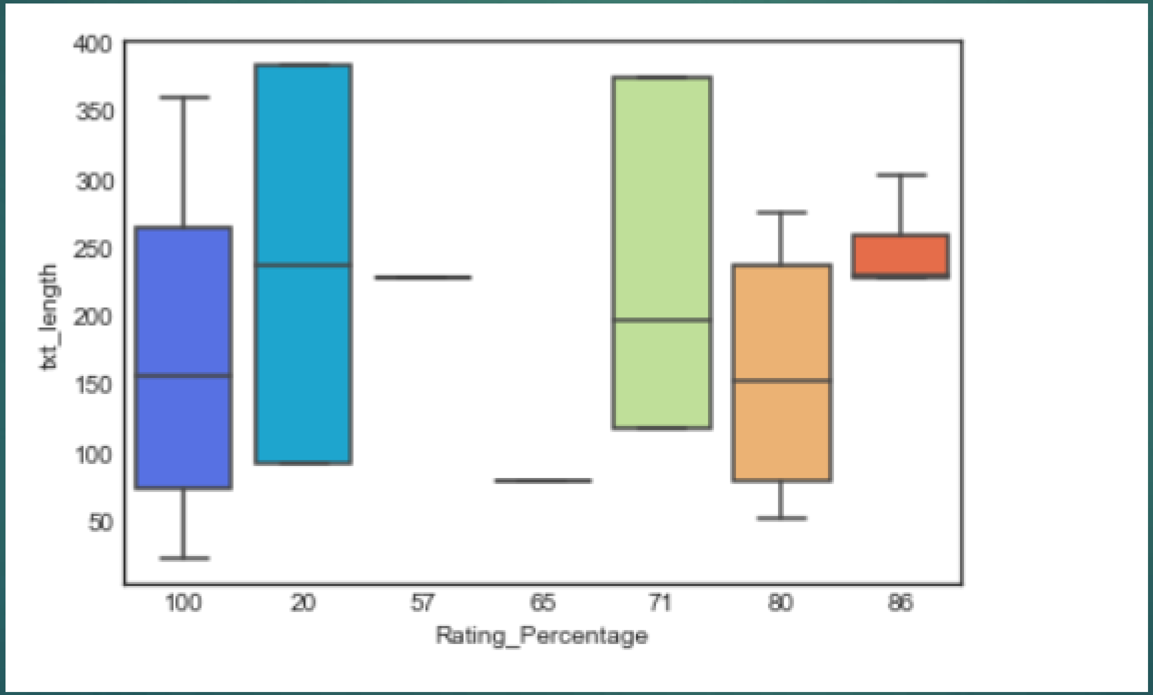
	Hotel_name	Review_Title	Review_Text	Sentiment	Rating_Percentage	txt_length
0	Accord Metropolitan	Excellent comfortableness during stay	Its really nice place to stay especially for b...	3	100	74
1	Accord Metropolitan	Not too comfortable	It seems that hotel does not check the basic a...	1	20	385
2	Accord Metropolitan	NaN	Worst hotel I have ever encountered. I will ne...	1	20	92

EDA

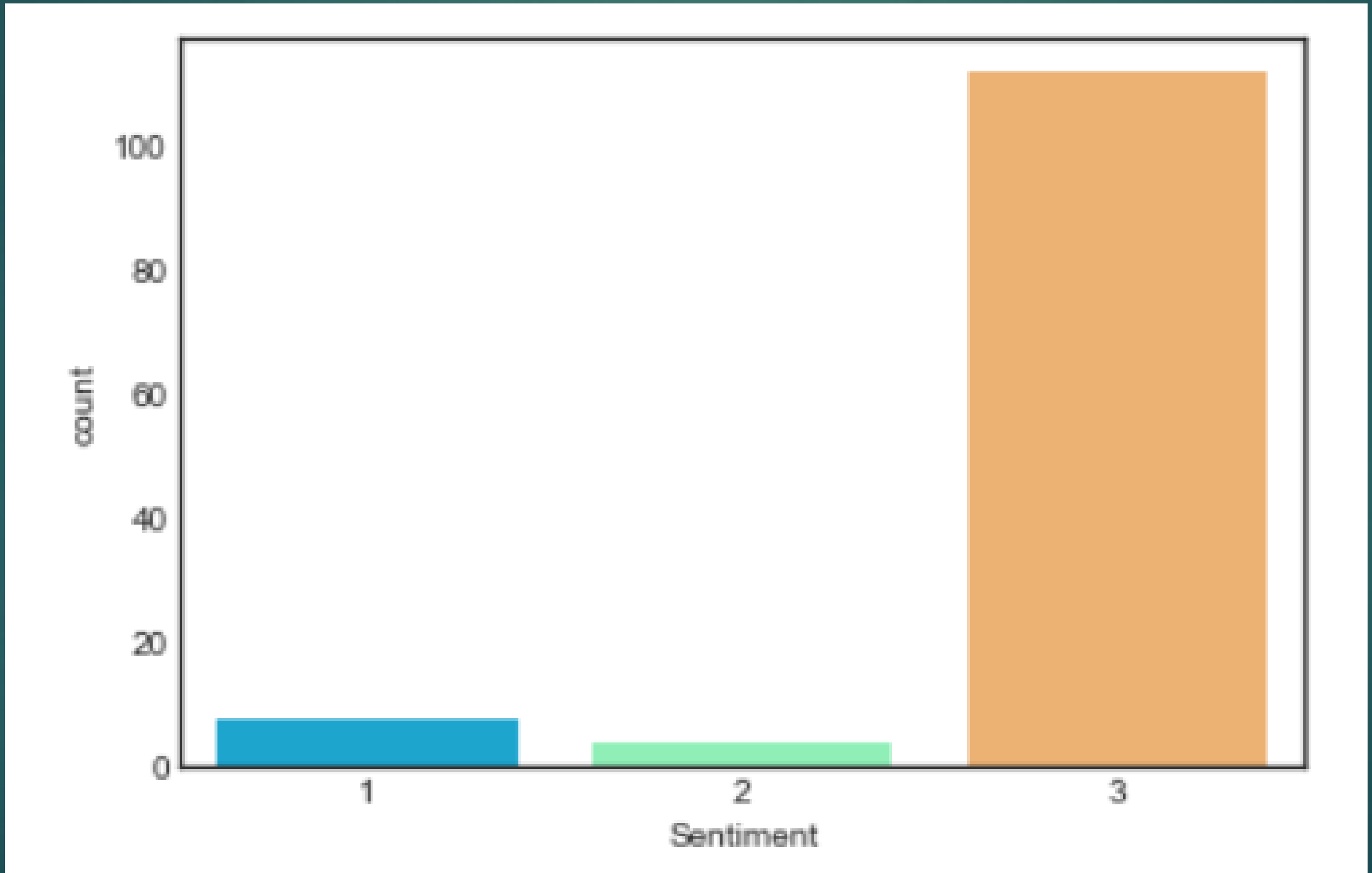
- Use FacetGrid from the seaborn library to create histograms of text length based off of the ratings.



Creating a boxplot of text length for rating



Creating a countplot of the number of occurrences



Sentiments Analysis – Scarping

- Steps Followed for cleanup
  - Clean up activities include processing stop words, removing repeating words and further cleanup of punctuations and meaningless words
  - Make sure that the stop words are working and prepare the methods for processing the stop words and punctuations. We'll also remove very common words, ('the', 'a', etc..).
  - **ext Pre-processing:** The classification algorithms need numerical feature vector in order to perform the classification task
  - Bag-of-Words (bw_trans) takes care of the words processing. It transforms the entire dataframe with a reduced memory space etc.

Sentiments Analysis – Scarping (2)

- **Text Normalization¶**
  - Continue normalizing text. Such as Stemming or distinguishing by part of speech.
- Vectorization
  - Now we need to convert messages to vector. We convert each message, represented as a list of tokens
  - Frequency of word occurances
  - Weight of the counts
  - Normalize the vectors to unit length

Term frequency-inverse document frequency(TF-IDF)

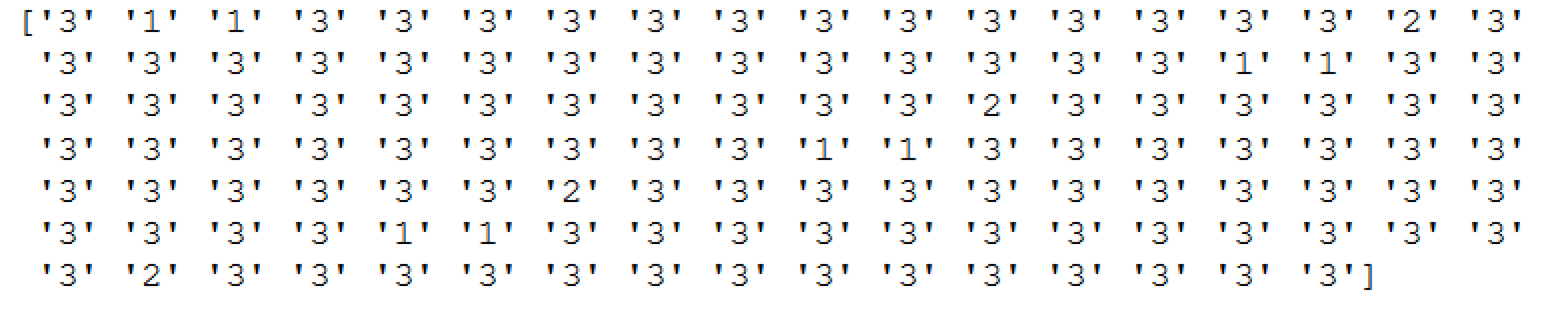
- TF-IDF is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus.
- Now use the SciKit learn to implement the TF-IDF logic in python

TFIDF implementation

- Steps
  - Check the vocabulary and see the importance of words
  - Check the important of some random words words
  - Transform bag-of-words corpus into TF-IDF

Training Model for Navie Bayes

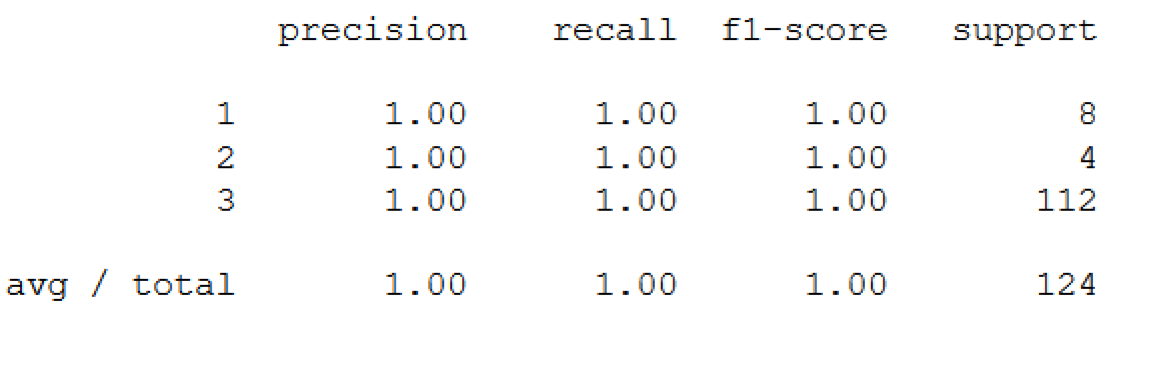
- Selecting the classification scheme and training
  - We'll be using scikit-learn Naive Bayes classifier in our scenario. The data is fit into the MultinomialNB class object
- **Try some prediction using sample data**
- Check the predicted values
  - Now have a quick check on the predicted values



```
['3' '1' '1' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '2' '3'
 '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '1' '1' '3' '3'
 '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '2' '3' '3' '3' '3' '3' '3'
 '3' '3' '3' '3' '3' '3' '3' '3' '3' '1' '1' '3' '3' '3' '3' '3' '3' '3'
 '3' '3' '3' '3' '3' '3' '2' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3'
 '3' '3' '3' '3' '1' '1' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3'
 '3' '2' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3' '3']
```

Text Analytics Model Evaluation

- Check the accuracy using the confusion matrix



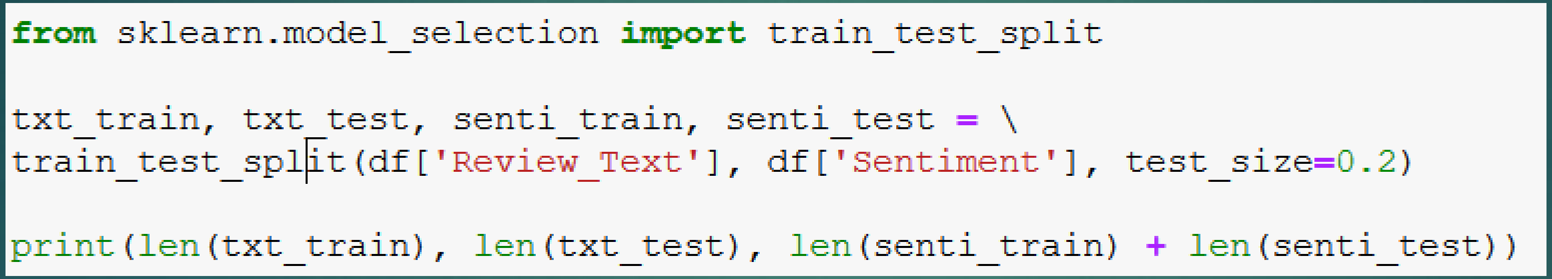
```
             precision    recall  f1-score   support

          1       1.00      1.00      1.00         8
          2       1.00      1.00      1.00         4
          3       1.00      1.00      1.00       112

avg / total       1.00      1.00      1.00       124
```

Train Test Split

- Train Test Split of original data for actual training of model



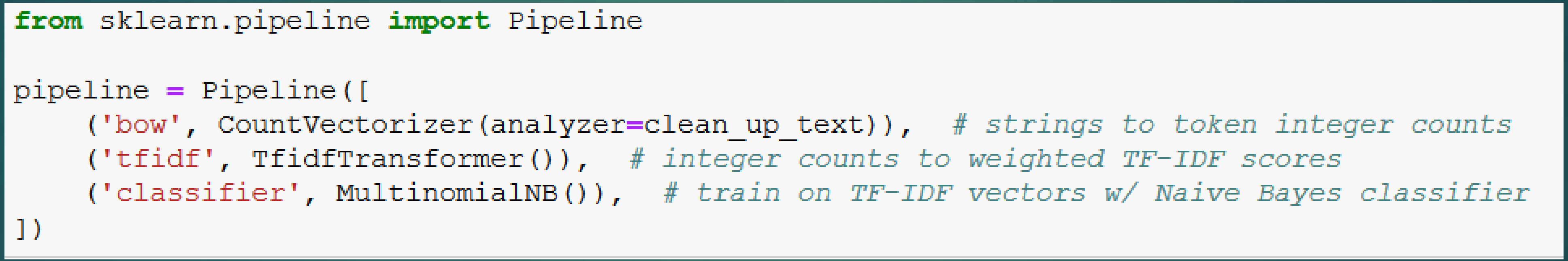
```
from sklearn.model_selection import train_test_split

txt_train, txt_test, senti_train, senti_test = \
train_test_split(df['Review_Text'], df['Sentiment'], test_size=0.2)

print(len(txt_train), len(txt_test), len(senti_train) + len(senti_test))
```

Creating a Data Pipeline

- We use SciKit Learn's pipeline capabilities to store a pipeline of workflow. This will allow us to set up all the transformations that we will do to the data for future use.



```
from sklearn.pipeline import Pipeline

pipeline = Pipeline([
    ('bow', CountVectorizer(analyzer=clean_up_text)),  # strings to token integer counts
    ('tfidf', TfidfTransformer()),  # integer counts to weighted TF-IDF scores
    ('classifier', MultinomialNB()),  # train on TF-IDF vectors w/ Naive Bayes classifier
])
```

Gather details on the hotel data¶

- Steps for data preparaton
  - **Get the city data¶**
  - **Get the city cordniates**
  - **Save City Data**
  - **Visualize the map data**
  - **Gather Foursquire API for live sentiments (Data Scarping to get the required data**
  - **Collect information for all the interested venues**

Gather Live Sentiments

- Now you gather sentiments for the intereted venues. IN our case we focus on the **Lulu Center** with Venue ID "*4d3057caa62d721ec754997d*" food center and see how the sentiments works for this food center
- Steps
  - We can iteratively call the details of the venue and collet the data into a DataFrame from JSON
  - Send the foursquare API to gather sentiments of single Venue¶
  - Tips (text) collected in a dataframe

Using the Sentiments Predictions¶

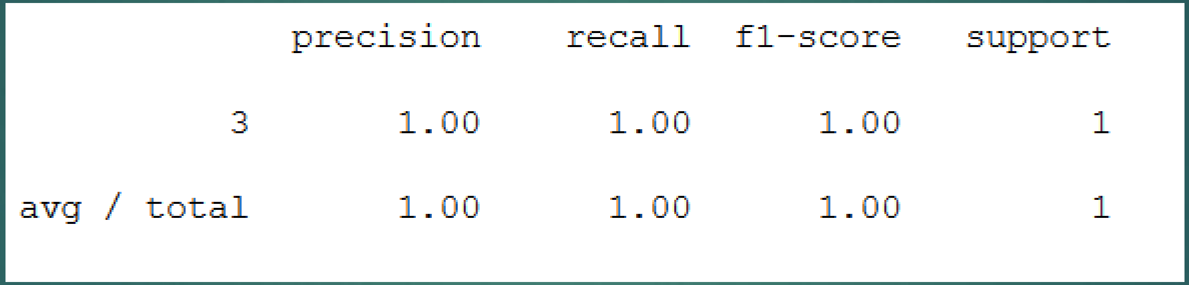
- Now this is the time to utilize the sentiments analytics model we created as a classification model based on Nave Bayes. We will now use the trained model based on the good quality dataset we got before.
- Optionally, we can gather more similar comments from Foursquare and update the training database

Prepare trainng set for Live Data

- Get an empty dataset and update the records. For testing purpose and simplicity, we have only one record to check
- **In our case it Predicted to have 3 as the sentiments ranking**
- **Check for accuracy for our result**

Compare the results

- There is only one record and there is no much meaning in checking the classification report.

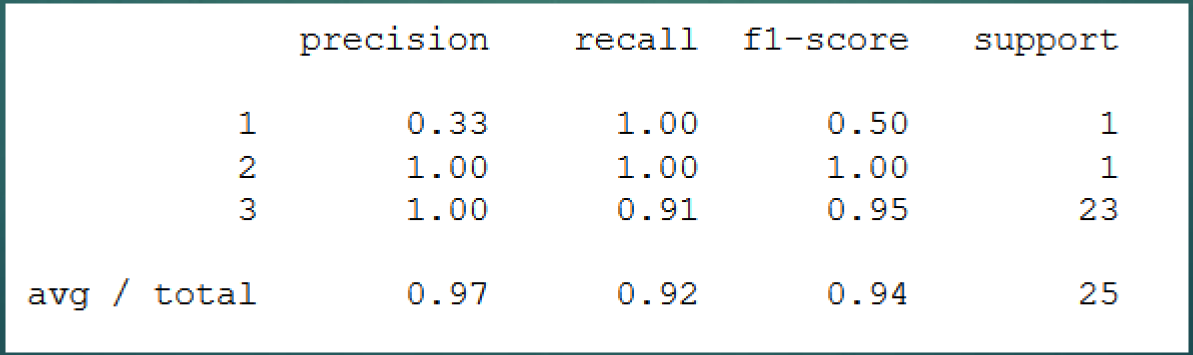


```
             precision    recall  f1-score   support

          3       1.00      1.00      1.00         1

avg / total       1.00      1.00      1.00         1
```

- Now check the result to the old result. It is found:



```
             precision    recall  f1-score   support

          1       0.33      1.00      0.50         1
          2       1.00      1.00      1.00         1
          3       1.00      0.91      0.95        23

avg / total       0.97      0.92      0.94        25
```

Final Remakrs¶

- The score says the quality of the model is good enough for this scenario. Further improvements to the model can be done based on more refied data and trying different algorithms with different hyperparameter